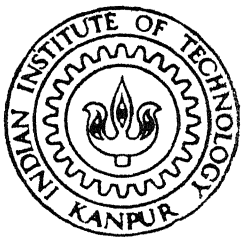
GESTURE BASED TELE-OPERATION

by

SAMBIT KUMAR DASH



DEPARTMENT OF MECHANICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

APRIL, 1998

GESTURE BASED TELE-OPERATION

A Thesis Submitted

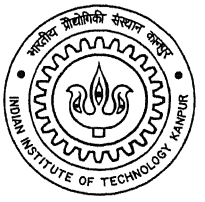
in Partial Fulfilment of the Requirements

for the Degree of

Master of Technology

by

SAMBIT KUMAR DASH



to the

DEPARTMENT OF MECHANICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

April, 1998

CERTIFICATE

It is certified that the work contained in the thesis entitled **GESTURE BASED TELE-OPERATION** by Mr. **Sambit Kumar Dash**, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Dr. A. Mukerjee
Associate Professor
Department of Mechanical Engineering
Indian Institute of Technology Kanpur
Kanpur, 208016

April, 1998

Acknowledgement

I am indebted to Dr. Amitabha Mukerjee for introducing me to the wonderful world of image processing. He has been constantly guiding and encouraging me in persuing this work. At the same time he has given me ample oppertunity to implement my own ideas and thus allowed me a lot of freedom in formulating the problem.

I am thankful to Prof. H. Hatwal for extending the facilities of the Center of Robotics for the completion of this work. It was a pleasure to be in the Center for Robotics and should acknowledge my gratitude to all the Staff of this center.

Mr. Susmit Sen has always been a constant source of encouragement. He would come up with a new gadget to give shape to my ideas. Of course this tryst of developing a complete system would ask a lot from him in the future.

I would like to thank my friends Atul Gupta, Debabrata Dash, Bikram Gupta, Vikas Gupta, S. Subramaniam, Abhishek Rawat and Mukesh Prasad Singh for their warm affections to make my stay at IIT Kanpur a pleasant one

Last but not the least I would like to express my gratefulness to all those who directly or indirectly helped me through the successful completion of my work. Partcularly, how can I forget Raju who has always been with me whenever I have needed him most.

My parents have always been the source of encouragement in my life.

Their patience and encouragement in crucial moments has given me the courage in completing this work

Sambit Kumar Dash
IIT Kanpur
April, 1998

Abstract

Teleoperation is traditionally done via joystick or keyboard from a dedicated control station. Gestures provide a rich intuitive communication interface for controlling real devices. Gesture recognition has been mostly based on specialised gesture pick up devices like datagloves or magnetic position sensing Polhemus. With the advent of better image capturing and processing systems these special gesture capturing devices can be replaced with non-intrusive image processing systems. This work is an attempt to integrate computer vision to the control of real devices. We develop two different gesture libraries in symbolic and pointing modes, and use them to control a manipulator (symbolic), control a mobile robot (symbolic) and a manipulator (static pointing gestures).

We observe that one of the strengths of symbolic gestures is that the same gesture set can be used both for a mobile robot and a manipulator. By combining spatial directives for incremental motion with symbolic instructions for controlling velocity, gestures can provide highly accurate control.

Contents

List of Figures

List of Tables

Chapter 1

Gestures as a Human Computer Interaction system

Recent advances in sensors and signal processing has enabled the technologies of voice, gesture and face recognition [16]. These will gradually enable a new model in human computer interaction (HCI). The keyboard and mouse may slowly give way to more human friendly interfaces.

Here we focus on gestures as a mechanism for communication. Unlike the 2-D mouse, gestures operate in a 3-D world and constitute a much richer communication interface. In addition to symbolic gestures such as 'bye bye' or 'come here', gestures can also be used to point and give spatial directives, a role with immense potential in robotics and other 3-D motion control. In this work, we develop a gesture operated system to control tele-operated robotic systems. Virtual Reality and tele-presence systems operate on a similar framework, but the major difference lies in their definition of the work environments. The VR environment is synthetic, machine generated while the tele-presence system is a model of an existing environment. This model is normally a video image of the operating area or can be sensory feedback of various parameters (like force feedback) in the operating environment.

1.0.1 Tele-operation

Tele meaning far, joins with operation to denote operation from far. Tele-

operation is widely used in robotic applications such as nuclear or space (hazardous), surgery (lack of access), or even for ease of use (control station).

Typically, tele-operation systems use dedicated communication channels to link up the control station to the work environment. Gestures are intuitive, widely used e.g. aircraft parking, EOT crane control, ship yards. They are mostly applicable for free motion and not compliant tasks. Gestures can be captured by dedicated hardware or by vision-based systems.

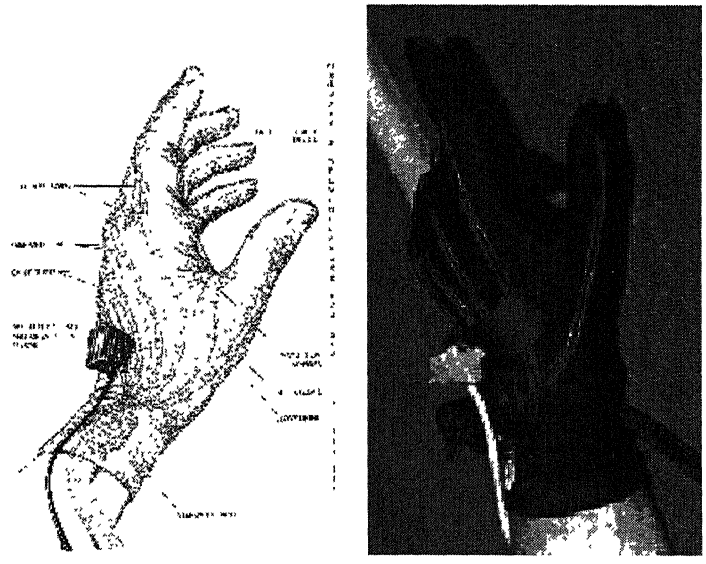
Earlier gesture operated systems have emphasised on special gesture capturing devices like DataGlove (Figure: 1.1)[1], Polhemus[2] 3-D position digitising products (Figure 1.2) etc. to calculate the finger orientation and hand position accurately. This method is quite cumbersome as calibration is to be carried out for each individual[3]. A good review of glove based input has been given in [23]. Image processing and computer vision gives an alternate approach in describing these gestures, which is less encumbering and non-intrusive. Use of passive sensors like sonic and infrared pulses can be intrusive in sensitive environments [4]. Special devices have another disadvantage in tele-operation environments. The remote user, transmitting the signals, may not have access to a DataGlove and its signal interpretation hardware If he/she is at a remote location, here vision holds an advantage. Since, images of the bare hand or inert glove can be easily captured and transmitted by increasingly available video conferencing systems.

Visual interpretation of gestures though non-intrusive suffers from var-

[1]shown in the figure are two different types of DataGloves. The first one called a Digital Data Entry Glove measured finger flexure, hand-orientation and wrist-position, and had tactile sensors at the fingertips Each of the sensors could be repositioned on the device for data entry needs The orientation of the hand was tracked by a video camera The second one is a DataGlove by VPL research, is a Lycra glove with optical fibers running up each finger The intensity of the light at the end of each finger increases The fiber monitors the bend of the first two joints of the finger, and the computer determines the relative flexure of each finger according to the intensity of the light generated by the optical fiber The point angles of each finger joint must be calibrated for each individual user in order to get accurate measures Excerpts from http://ils.unc edu/alternative/dataglove.html

[2]It's a company working in the developing equipments for measuring the 3-D position in a given region using electro-magnetic techniques

[3]from http.//ils.unc.edu/alternative/dataglove.html



Digital Data Entry Glove A VPL Data Glove

Figure 1.1: *DataGlove:* The first glove called a Digital Data Entry Glove. The second glove is a DataGlove, developed by VPL research. These are devices used for simple gesture recognition and general tracking of hand orientation. It is not able to process complex gestures or fine manipulation, and is not able to track quick motions. (figures and information from `http://ils.unc.edu/ alternative/ dataglove.html`)

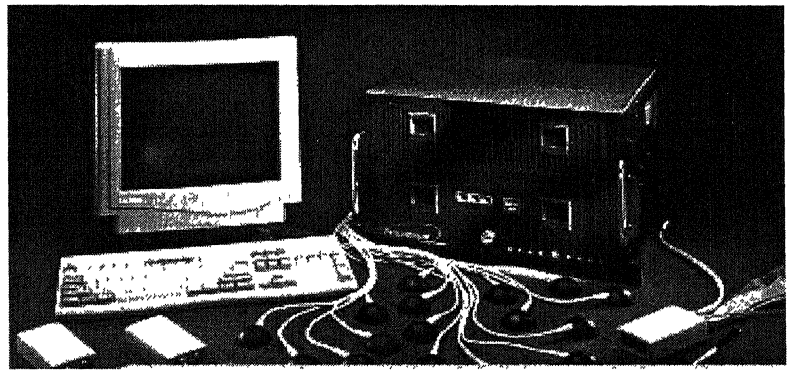


Figure 1.2: *Star Track:* Motion Capture Server – Rack mounted electronics for computing position and orientation up to 32 receivers and up to 2 TRACK*BELTs. TRACK*BELT is a wide belt containing electronics up to 16 receivers. Capture Space Area is 25' x 25'. All information and figures from `http://www.polhemus.com/`.

ious inaccuracies. The inaccuracies can be noise in capturing the image or can be noise in identification of the gesture. The noise in image capturing is device dependent and can be improved upon by using better cameras, but the second kind of noise is a major bottleneck in all vision based systems. Most gesture operate systems work on heuristics or principles of human behaviour, which fail drastically with the presence of surrounding objects or relative movement between them. Many a times a controlled environment is created using special coloured backgrounds, clothing or gloves for the ease of understanding of gestures. Special conditions have been used by various authors like use of gloves, dark clothing and dark background [10, 11, 12]. Some have reported a reduction in accuracy of recognition of gestures by not using gloves. However, these accessories are cheap and can give results with significantly improved accuracy. Transmission and understanding of images are getting more and more feasible with current computational power. These vision based systems are interactive, hence can be used to control online devices.

1.1 Gesture and Interpretation

1.1.1 Definition

Gestures can be defined as – "expressive movement of a part of the body especially hand or head". We will focus only on hand gestures in this work. Head gestures are quite subtle and involve tracking the eyes, lips, eyebrows, cheeks etc. when accompanied by hand movements, they are fairly complex to analyse. In any event motion directives are almost always arm/hand gestures. The gestures depend on intention and mode of communication. Figure 1.3 outlines the production and interpretation of gestures.

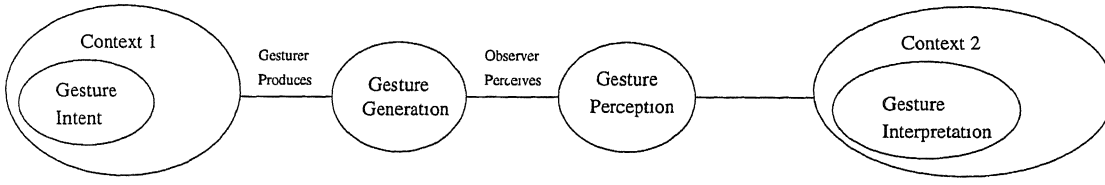


Figure 1.3: *Production and Interpretation of Gestures:* Gestures are the outcome of the intention of the operator. A proper understanding of the gestures depends on the translation of the generated gestures by the perceiver in the context in which it is generated.

Context Dependency

For gestures to be understood, it is essential that there should be a degree of similarity between the context in which a gesture is generated and the context in which it is perceived. This is mostly achieved in normal communication where the gesture maker (speaker) and the gesture perceiver share the same physical space and often the same sense of task or purpose. In our application of tele-operation, the physical space is very different for the gesture maker and gesture perceiver and therefore, the range of ideas that can be expressed is limited. One such important example can be 'pointing at' and 'pointing in'. Normally, pointing is always carried out within certain contextual frameworks. But, unless the context is not properly specified the pointing normally means the object to which the gesture operator is pointing at. Again, pointing can also refer to the direction to which the gesture operator is pointing in. The difference in such gesture interpretation is described in section 1.4 3 in greater details. Note that a gestural mode requires prior context definition, e.g. 'Look up there' before physically pointing with the arm or face. When we use pointing in this work, it is assumed that this contextual framework has already been established.

Generation and Perception of Gestures

Generation of gesture is context dependent. Any arbitrary hand movement cannot be described as a gesture. Any such movement of the hand is called

noise in gesture understanding Elimination of such noise from effective gestures has always been challenging. Perception of various gestures demand a a better understanding of visual gestures. Hence there is a need to classify the gestures Figure 1 4 gives a broad classification of gestures with reference to the intention of the gesture. The other classification can be based on physical motions of the hands or can depend on the different levels of human anatomy This classification is shown in figure 1.5.

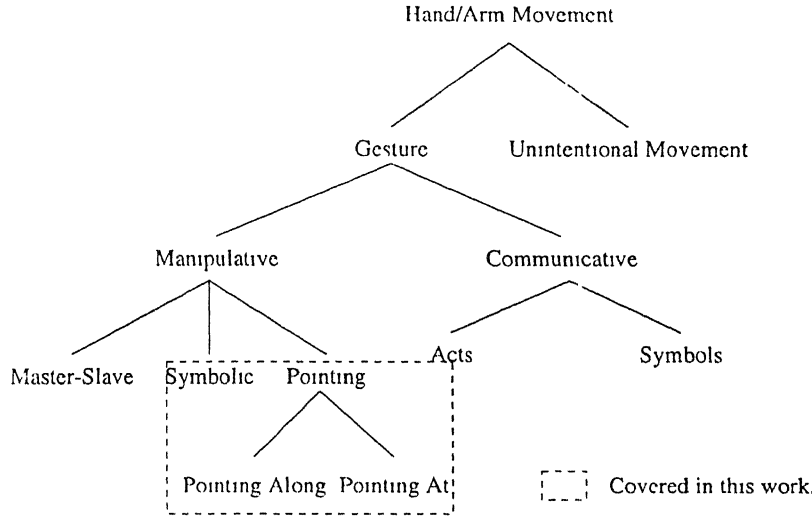


Figure 1.4: *Classification of Hand and Arm Movements:* In this work, the main focus has been on manipulative gestures Actually in a sense some degree of manipulation is also achieved with pointing and symbolic gestures in case of an anthropomorphic robot. Symbols like Rotate CW, Rotate CCW are a few commands which can be used to rotate the wrist joints of the manipulator - extension of taxonomy described in [16].

1.2 Building models for Gestures

The taxonomy of gestures as shown in figure 1.4 and 1.5 has led to different authors modelling gestures in various ways. A brief summary of their work is presented in table 1.1 The models can be either static or dynamic depending on the kind of gestures that are to be studied.

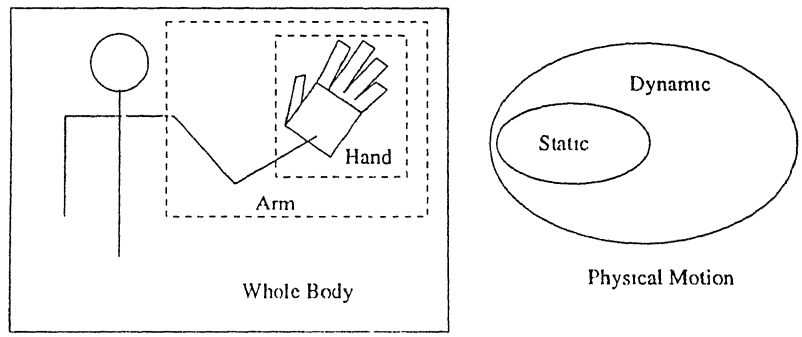


Figure 1.5. *Classification of Gestures based on utilisation of levels of human anatomy*A gesture may involve only the hand ("Thumbs Up"), arm ("Bye-Bye") or the whole body ("bow") Temporal distinctions, figure (b) relate to whether a gesture has a strong temporal component (dynamic) or little or no temporal variation (static).

Static Vs Dynamic Gestures: Gestures constitute a preparation phase, a stroke phase and a retraction phase [16, 25]. A gesture is said to be static if the stroke phase involves little or no motion. The static gestures are easy to incorporate, require lesser amount of processing and hence can easily be incorporated on relatively slower communication setups in a off-site tele-operation. Most gestures are dynamic in nature. Even in case of static gestures, the preparation and the retraction phases are dynamic in nature An inherent assumption in study of static gesture is that no stroke or retraction phase occurs during capturing the gestures

1.3 Gesture Recognition

The gesture recognition algorithm is directly dependent on type of model used for parameter extraction. Some popular strategies in gesture recognition are based on the study of human body (parts) kinematics [18, 19] by comparing with a given CAD model, tracking of human model based on 2-D point sets derived from image sequences [26] or dynamic time warping using correlation based view models [5] or using Hidden Markov Mod-

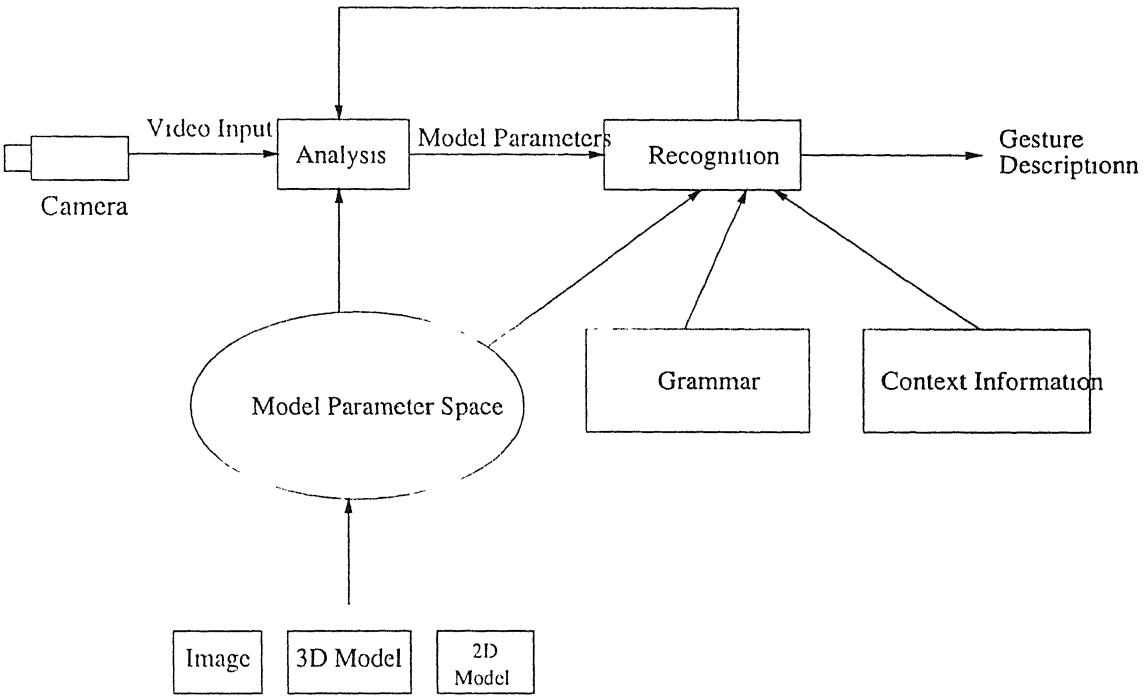


Figure 1.6 *Visual Gesture Recognition*: The image input is compared with the gesture model which can be a CAD model (2-D or 3-D) or an image. This comparision gives the gestures parameters as the output. The recognition of the image model is carried out on the basis of a gesture grammar and the context information.

els [2, 20, 21, 22, 25]

CAD models Vs. Image models. Gestures are represented by comparing the image with the parameters of a given model. These models are either image based or CAD-based. The visual gesture recognition paradigm is shown in figure 1.6. CAD based models for gesture recognition have been used to recognise fine details of hand configuration [18, 19]. Similarly, P-finder [26] uses a 2-D contour shape of a human being to learn how to the track the human in real time. In this work, we compute the parameters from the gestures without comparing the input gesture every time with a model. The final gesture is recognised by comparing the computed parameters with set parameter values computed from experiments conducted earlier.

who	temporal	intent	body
Brand, Oliver, Pentland [2]	dynamic	Communicative Symbol	Whole Body
Wilson, Bobick [25]	dynamic	Modalizing Gesture & Pointing	Mouse, Magnetic spatial tracker, Whole Body
Brand, Essa [1]	dynamic	Act	Gross Motion
Rehg, Kanade [18, 19]	static	Manipulative	Fine Motion
Wilson, Bobick [24]	dynamic	Manipulative	Fine Motion
Mukerjee, Dash [12]	dynamic	Manipulative	Hand Gestures
Darell, Pentland [5]	dynamic	Communicative Symbolic	Whole Body
Hunter, Schlenzig, Jam [6]	dynamic	Communicative Symbolic	Arm Gestures
Katkare, Hunter et. al [7]	static	Manipulative Symbolic	Arm & Finger Gesture
Mishra, Singh et. al. [10, 11]	static, dynamic	Pointing, Symbolic (Manipulative)	Arm Gestures
Starner, Pentland [21, 22]	dynamic	Communicative Symbolic	Finger Gestures
Pook, Ballard [17]	static	Pointing	Arm Gesture

Table 1.1: *Experiments in gestures – Classified*

1.4 Gesticulation to Manipulation

A broad taxonomy of gestures are described in figure 1.4. Our focus is to control real devices using gestures. Hence, the major emphasis on manipulative gestures. The manipulative gestures are primarily classified as:

- master-slave
- symbolic
- pointing
  - Pointing Along a direction
  - Pointing At an Object

1.4.1 Master-Slave

As the name suggests, here the device which is to be controlled is used as a slave to the operators gestures i.e. it exactly follows the master's movements. The inherent demerit in these systems is that the master and slave must have similar configuration. A human arm can control some joints of an anthropomorphic robotic arm [10], but the same system cannot be used for a non-anthropomorphic system like a mobile or flying robot. Even while controlling an articulated arm, the robot's link parameter's may not be proportionate to a person's arm, resulting in kinematic mismatch such as singularities, and undesired end effector orientation.

1.4.2 Symbolic Gestures

Symbolic gestures such as "Come here" are commonly used in human communication These gestures are intuitive and have a strong cultural component, e.g. "he is nuts" gesture pointing at the ear. Hence these are stable, easy to use and a more humanlike means of communication. Most symbolic gestures are dynamic in nature. But, they have static variations which

are easy to recognise and a interpretation can be carried out at low computational cost. Gestures have both spatial as well as temporal attributes attached them. Our analysis is focused on the spatial aspect and temporal variation is not the focus of our analysis.

1.4.3 Pointing Gestures

Pointing gestures are commonly used in human interaction. Normal acts involve some amount of basic pointing action mostly as an emphasis. Pointing is not just limited to hand motions even eye balls and head movements are used as pointing devices. Pointing is a directional directive and without a context has a limited application. For example, teacher pointing towards a blackboard need not essentially mean a particular location on the blackboard. It can be just to bring attention to the blackboard as a whole. But, such actions are accompanied with speech which gives a focus to the pointing action. In dealing purely with gestures, gestures can be translated as point-at or point-along:

Pointing At an Object

Pointing at is a common phenomenon. But, this involves the objects location with reference to the gesture operator as well as the observer's position. For example, a command "move to a particular object" requires the knowledge of the objects exact position with reference to the robot. In such circumstances, one solution can be to search a complete map of the work area. But, such information loses its importance as soon as new objects are introduced to the system. The other feasible solution is to maintain a measure of the robot's relative position with respect to the operator. But, the final question that remains is how to recognise the object. If the object shape etc. are known then object can be recognised using template matching. Any general pointing gesture precludes the use of a template. Hence, the nearest obstacle in the direction of the motion of the robot in the pointing direction can be

assumed as the pointed at object.

Pointing in a Direction

Here the observer is supposed to consider the direction of the pointing as generated by the gesture as the direction of pointing. It is similar to pointing to an object in the horizon. In fact in most computer games use a similar pointing strategy in utilisation of the pointing devices. Virtual Reality and Tele-operation systems insist on such pointing so as to give the operator the feel of the work environment. And the input given to the operator is exactly what is observed by the robotic device inside the work environment. One major advantage of this system is its ease in implementation, yet in a work space where both the gesture operator and the observer are present this kind of a strategy is quite confusing as the gesturer can see the operator's movement. In this work, we also follow the same convention.

1.5 Off-site Tele-operation

Most tele-operated systems are normally controlled from a control room which is near the robot's operation and thus the robot can be controlled from the control room through teach pendants and joysticks. These devices encumber the operator and sufficient operator training is required to effective utilisation of these equipments. Off-site tele-operation precludes use of such devices and replaces them with gestures. The gesture commands can be sent as unprocessed images by by a high bandwidth (e.g. ISDN) line. The processing of the image can be carried out in the control station of the robot and images of gestures generated in a remote site can be translated as gesture commands in the local cell. The images of the position of the robot can be sent to the remote site as a feedback to the operator. Thus, availability of a two way ISDN communication system can enable an operator controlling a real device in a remote location. A figure outlining this paradigm is given

in figure 1.7.

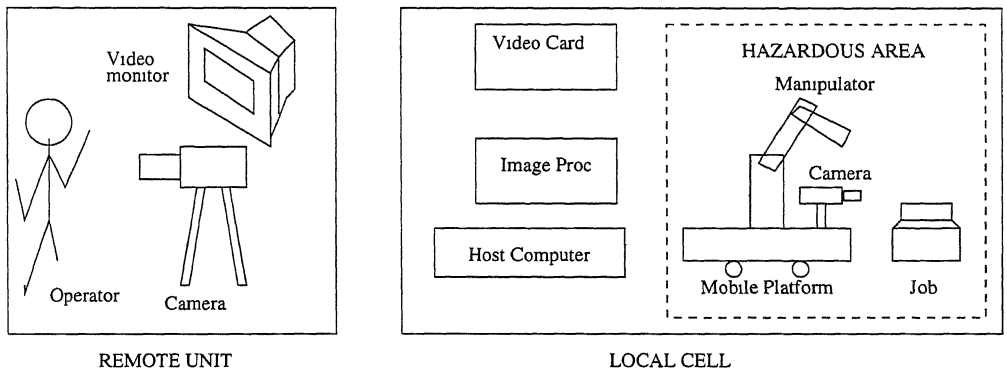


Figure 1.7: *Gesture video conferencing based teleoperation setup*

Chapter 2

Gesture Recognition & Applications

In this chapter we develop the models for the symbolic and pointing modes of real device control and show the outcomes of the experiments conducted on various robots, both mobile and manipulators.

2.1 Setup

The image processing equipment used is shown in figure 2.1.

The most common steps used in all image related experiments conducted in this work include:

- *Image Grabbing:* This includes capturing the digital image using the digital CCD camera After the image is grabbed it is stored in a buffer for further processing to be carried out.

- *Binarisation:* This is used to distinguish foreground from background and in our experiment the background was kept dark. Hence, the intensity of the background was below certain threshold value. In binarising, the pixels having intensities greater than that of the threshold value are assigned the maximum intensity while those having lesser intensity than the threshold value are assigned the minimum pixel intensity.

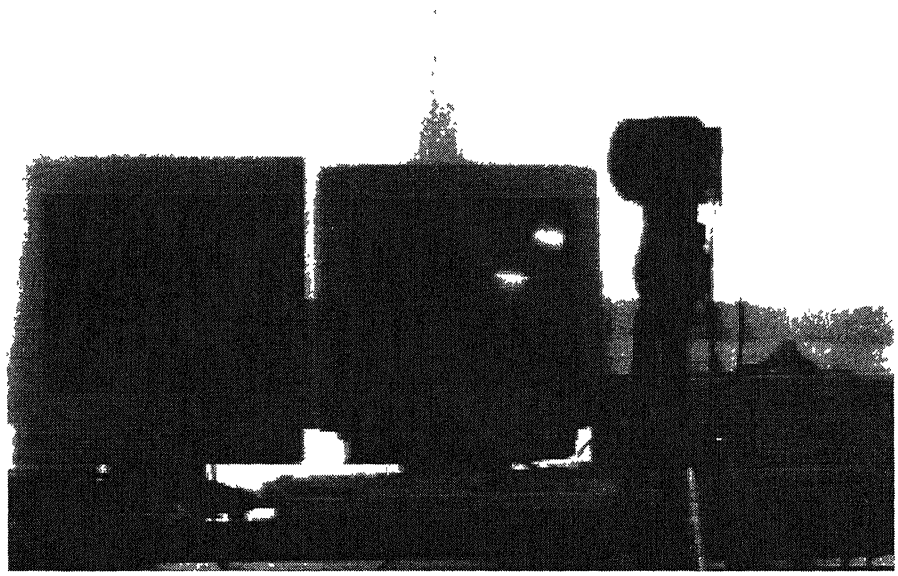


Figure 2.1: *Image Processing Equipment:* The image processing equipment consists of - *Camera:* Watec colour camera, with auto iris, auto focus lens, *Image Processing System:* A Matrox Image-LC card with ability to process grey scale images, and *PC:* A Intel-486 DX2, 66 MHz machine

- *Blob Analysis:* Blobs are group of pixels connected to each other. In a binary image there are two kinds of connectivity which are mostly used called the 8-connectivity or 4-connectivity shown in figure 2.2. Once the images are separated into blobs the shape properties of blobs like the center of gravity, the bounding box, moment of inertia can be found out.

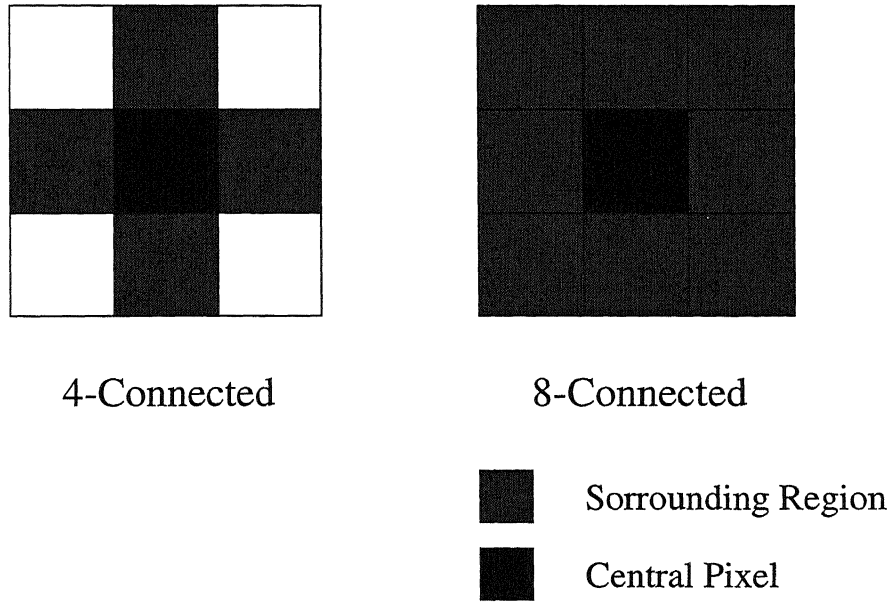


Figure 2 2: *Blob connectivity:* If the there exists a pixel of same intensity as the original pixel considered then the connected pixel is assumed to be in the same blob as the central pixel.

Gestures can be distinguished either by their position in space or from the shape of the hand (palm). These variation in space and shape can simultaneously occur in the images.The objective of the vision based gesture recognition is to follow the temporal variations in the shape / position of the hand. In th following sections we distinguish two types of gestures - symbolic and pointing.

2.2 Symbolic Gestures

Our work on gestures is motivated by the desire to control real devices such as manipulators or mobile robots. In the symbolic mode, a minimal gesture set may consist of the seven gestures: Move Left, Move Right, Move Forward, Go Back, Go Up, Go Down and Stop. Since this set of gestures is so limited we implement our system by independently dealing with both position and shape aspects of gestures. The gestures used in this work are static gestures i.e. the temporal variability is less important. We have initially created a set of gestures which depend only on the gross motions of the arm, we call it *Position Dependent Gestures* and we build another set of gestures which depend only on the orientation of the palm,we call it *Shape Dependent Gestures.*

Position Dependent Gestures

The seven elements of our gesture set are shown in figure 2.3 considering only position of the palm in the image region. One of the processed images of Go Up gesture is shown in figure 2.5.

Shape Dependent Gestures

Similarly, the gesture set shown in figure 2.4 are distinguished only on the basis of the orientation of the hand.

2.2.1 Gesture Interpretation

Position Dependent Gestures

The operator wears a black coat and stands in front of a black background. In case of position based recognition he wears white gloves which appears as two white blobs in the binarised image. The image space is broken into a 3×3 array. Depending on the position of the center of gravity of the hand blobs the gesture is interpreted. In this version also to distinguish the

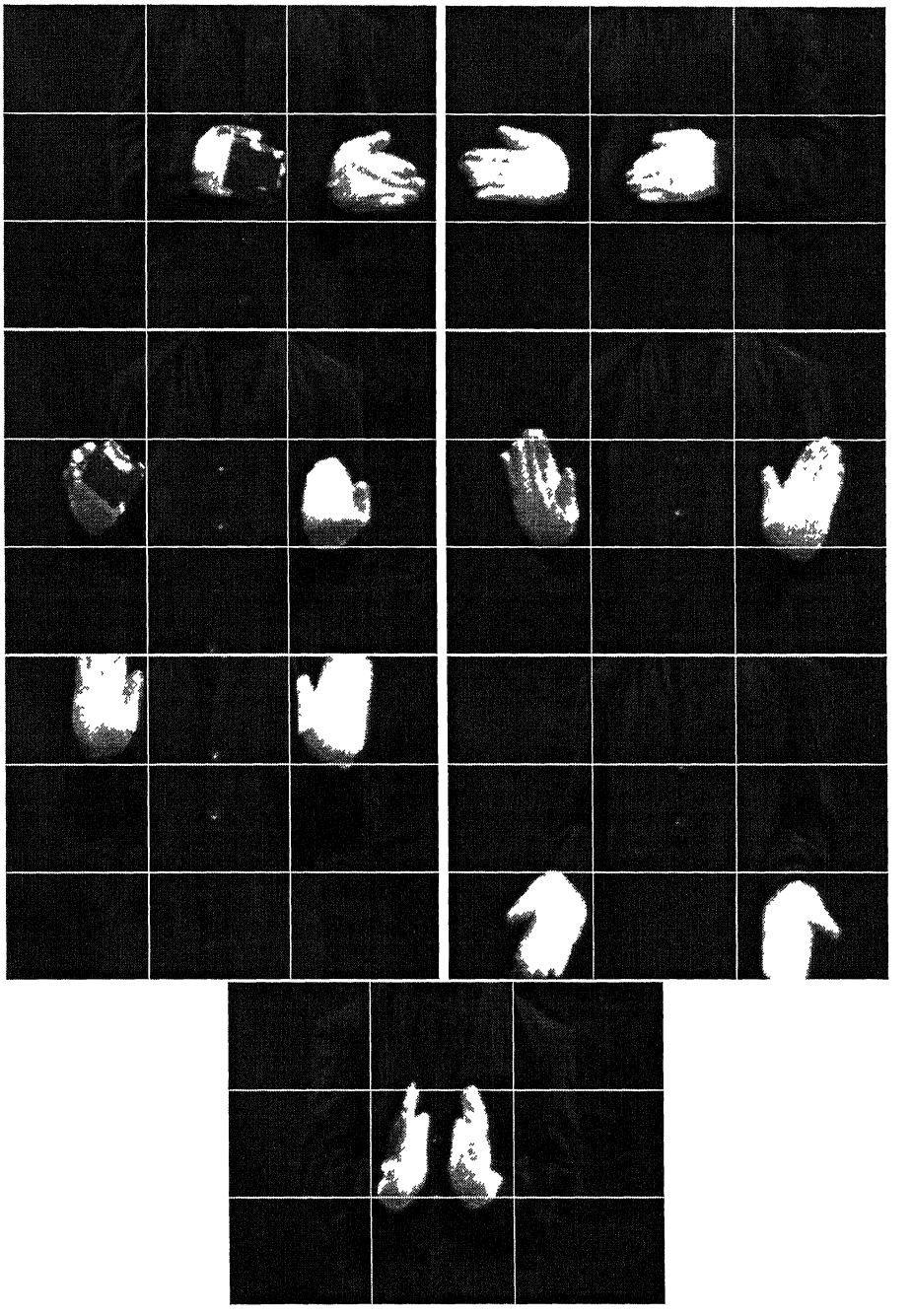


Figure 2.3: *Symbolic Gestures - Hand Position Dependent:* Top Row: Move Left, Move Right, Second Row: Move Forward and Go Back, Third Row: Go Up and Go Down, Bottom Row: Stop. Visual processing distinguishes the gestures based on the 3×3 regions as shown. The operator wears dark clothes and light-coloured gloves have been used to minimize effects of varied lighting.

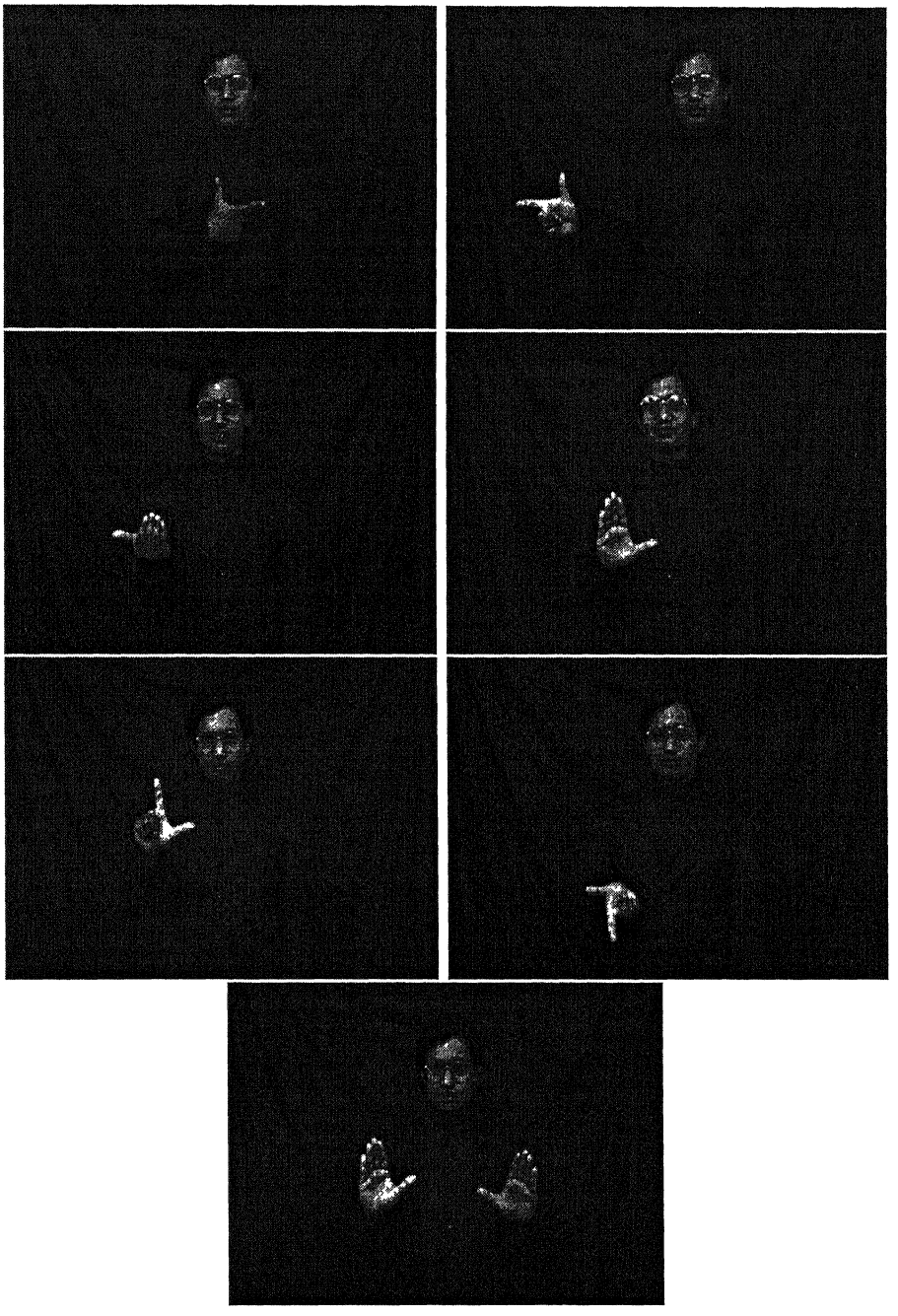
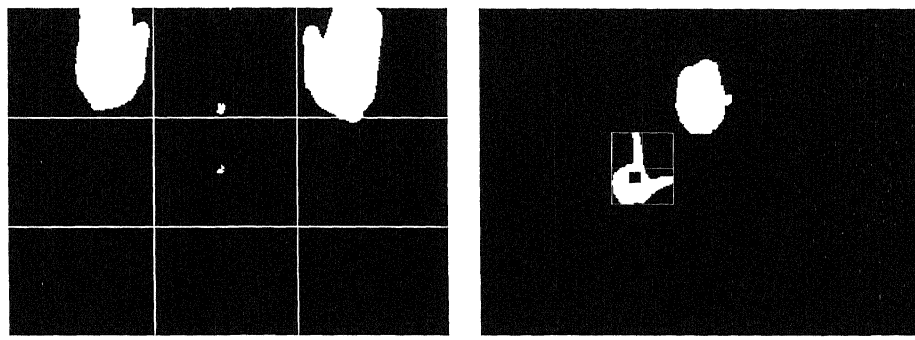


Figure 2.4: *Symbolic gesture set - shape based:* Top Row: Move Left, Move Right, Second Row: Move Forward and Go Back, Third Row: Go Up and Go Down, Bottom Row: Stop. The operator uses a dark clothing and stands in front of a dark background, and this helps in capturing the operators palm orientation. These gestures are based on the shape of the palm only. Visual interpretation templates for these gestures are shown in figure 2.6.



(a) (b)

Figure 2.5: *Processed Images:* Move Up gesture is considered from both (a) Position Dependent Gestures (b) Shape Dependent Gestures after processing. (a) Center of Gravity of both hands are in the upper blocks of the 3×3 grid. (b) The black patch in the hand blob is the location of the Center of Gravity. The position of this is towards the bottom left quarter of the hand blob enclosing box. The compactness factors of the hand and head blobs are 2.907 and 1.577 respectively.

forward gesture from the go back gesture a black patch is put on the back of the gloves. If the area of one blob becomes 60% of the other, then it can be assumed that the camera is looking at the back of the hand.

Shape Dependent Gestures

In the second set of gestures the user does not wear gloves, but continues to wear the black coat and stands in front of a black background for the sake of binarising and blob analysis of the image. Processed image of on of the gestures is shown in 2.5. The details of the recognition steps are explained in figure 2.6, where the orientation of the hand is determined from the Center of Gravity and compactness factor. The compactness factor is a measure of closeness to a circle for a given shape defined by:

$$\text{Compactness Factor} = \frac{\text{Perimeter}^2}{4\pi\text{Area}}$$

This factor is smallest for a circular shape i.e. 1.0 and greater than 1.0 for any other shape. Hence this becomes one of primary criteria of distinguishing the hand from the head as only area is not sufficient in deciding this.

As the camera comes close enough to the operator the hand comes closer to the camera than the face, hence differential scaling occurs in the two blob sizes.

2.3 Pointing Gestures

The pointing gestures are based on the user's hand orientation in 3-D captured by a single camera which sees the user from the front. Here, the operator stands in front of a black background and wears light coloured clothes (to distinguish the user from the background). Since the camera is viewing the user from only one particular angle and the need is to estimate the 3-D orientation of the hand, true length of the hand is needed as an input. But, in an off-site tele-operation the availability of any such measuring equipment is highly unlikely. So, in this case we have required the user to execute an initial pose with the hand; the user stands in front of the camera with his hands stretched outwards as shown in the top image of figure 2.8. The processing on the image comprises histogram-equalisation, binarisation followed by erosion and dilation. The final image has the user as a single blob. The top point of this blob is the head location, the vertical projection gives the position of the shoulder. The extreme left point of the blob gives the hand tip location. But the vertical projection as a measure of shoulder position is not complete, as a loose shirt can give a wider shoulder width. To do away with such problems the hand length is measured from the initial position by finding the distance from the head blob to the hand tip. The actual shoulder length is obtained as a ratio to the distance of the hand tip to head top point distance. An average measure of limbs of a human being to the height of a man has been shown in figure 2.7. Once the shoulder position is known, the subsequent images are used to calculate the hand tip position. From the true length of the hand (in the image coordinate frame) and the projected length of the hand the depth information can be

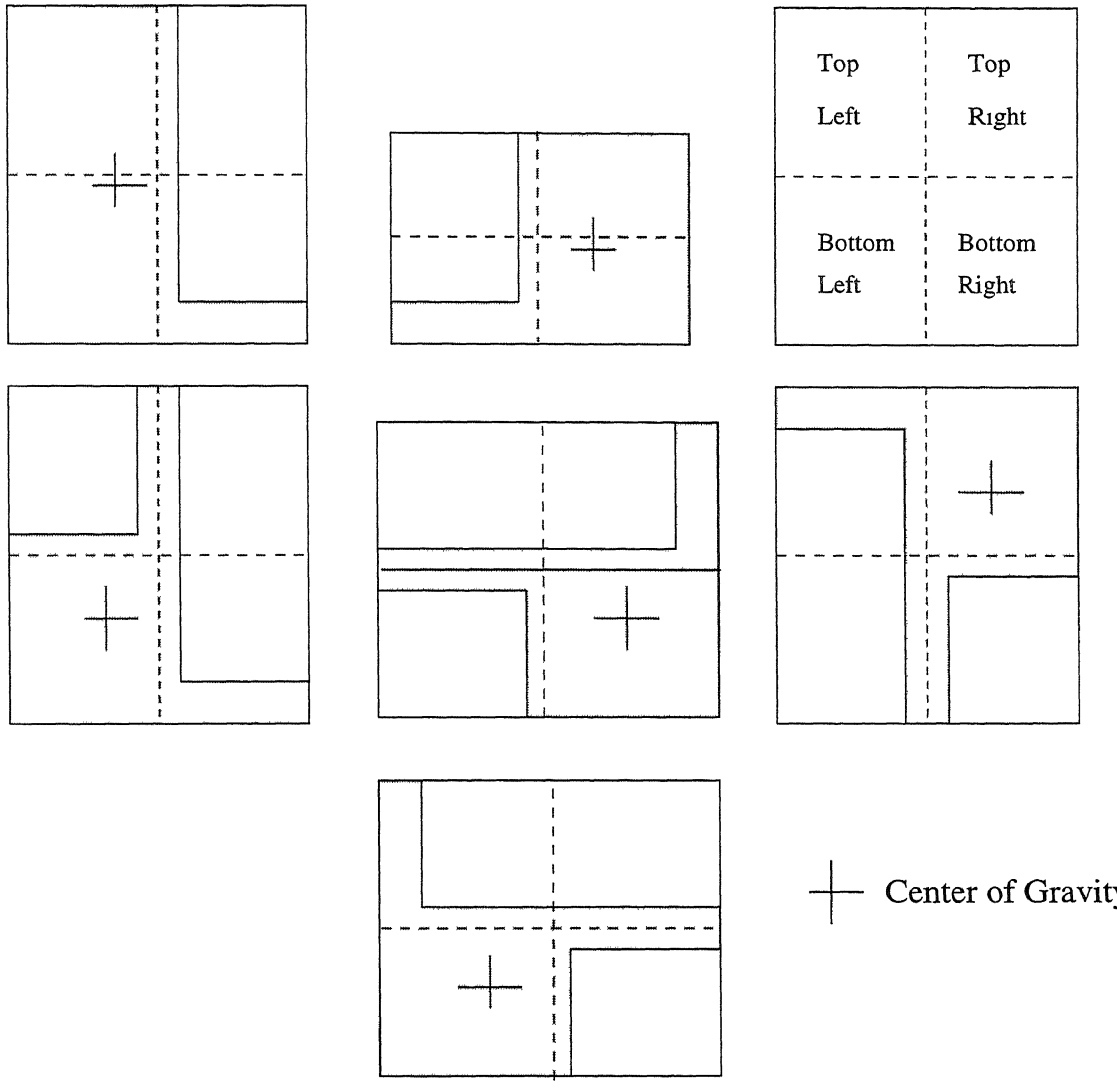


Figure 2.6: *Determination of Gestures from hand orientation*: The gestures from top left Go Back, Come Forward, Go Up, Go right, Go down, Go left. The Go back and Come forward gestures have a smaller compactness factor (less than a set threshold), hence can be distinguished from the rest of the gestures. Between these two gestures the CG gives a distinctive measure of the hand orientation. The Move up, Go down, Go right, Go left have larger compactness factor (beyond a set threshold). The CGs are a good measure to distinguish right, down and left & up gestures. The CG positions of the go up and go left are conflicting which is resolved with the position of the top left point of the hand blobs.

estimated, thus the pointing direction can be found out

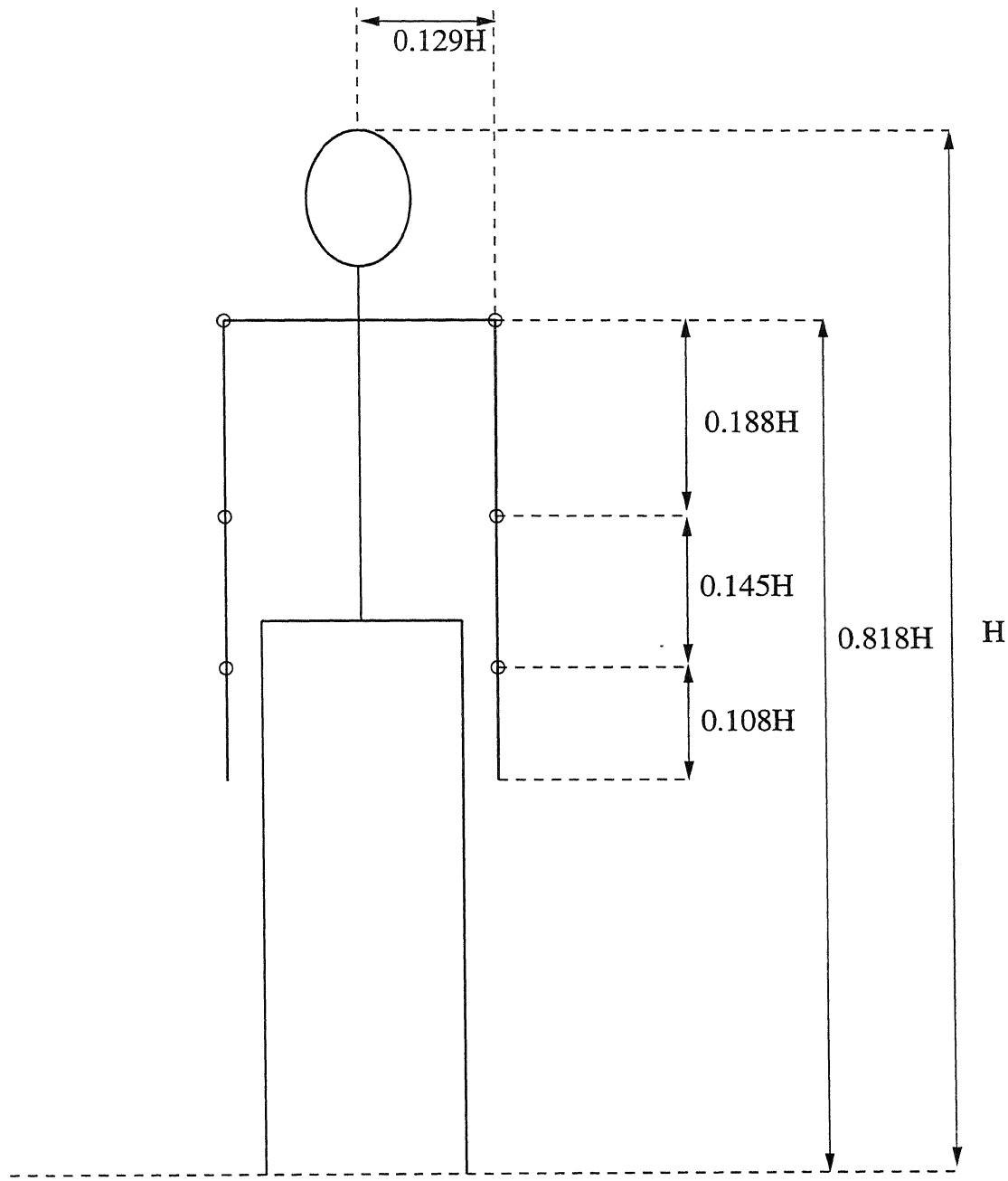


Figure 2.7: *Statistical measure of a man's anatomy:* All measures are carried out w.r.t the user's height H. Figure taken in parts from [14]

Figure 2.8: *Pointing:* The first figure shows the initial state of the operator for calibration. The second figure is any later figure where the actual pointing direction is estimated.

2.3.1 Sensitivity Analysis

Images are planar, while motion is cylindrical, hence variations image space are higher in some parts of the motion space though the motions may be equal. In our case, we are estimating the direction of pointing from the image space. Hence, the inputs are the x and y coordinates. Hence, sensitivity is dependent on the variation of subtended solid angle $\delta\psi$ as shown in figure 2.9.

From figure 2.9 using the rectangular to polar coordinate transformations.

$$\left.\begin{array}{rcl} x & = & \left(\frac{f}{f+d}\right) L\cos\phi\cos\theta \\ y & = & \left(\frac{f}{f+d}\right) L\sin\phi \end{array}\right\} \tag{2.1}$$

where f is the distance from the screen to the lens and d is the distance of the operator to the camera lens and L is the actual hand length.

The variations in the x and y coordinates w.r.t. the ϕ and θ can be written as:

$$\delta\mathbf{X} = [J]\,\delta\mathbf{\Theta} \tag{2.2}$$

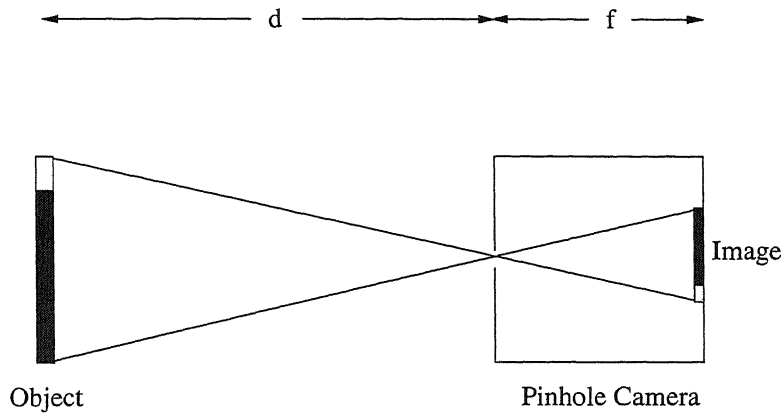
where $\mathbf{X} = \{\delta x \quad \delta y\}^T$, $\mathbf{\Theta} = \{\delta\phi \quad \delta\theta\}^T$ and $[J]$ is the Jacobian matrix denoted by:

$$\begin{bmatrix} \frac{\partial x}{\partial\phi} & \frac{\partial x}{\partial\theta} \\ \frac{\partial y}{\partial\phi} & \frac{\partial y}{\partial\theta} \end{bmatrix}$$

$$\left.\begin{array}{rcl} \frac{\partial x}{\partial\phi} & = & -L'\sin\phi\cos\theta \\ \frac{\partial x}{\partial\theta} & = & -L'\cos\phi\sin\theta \\ \frac{\partial y}{\partial\phi} & = & L'\cos\phi \\ \frac{\partial y}{\partial\theta} & = & 0 \end{array}\right\} \tag{2.3}$$

where L' = $fL/(f+d)$ which is the true length of the arm in image coordinate frame.

Taking the Jacobian inverse and computing the $\delta\mathbf{\Theta}$ the output is:



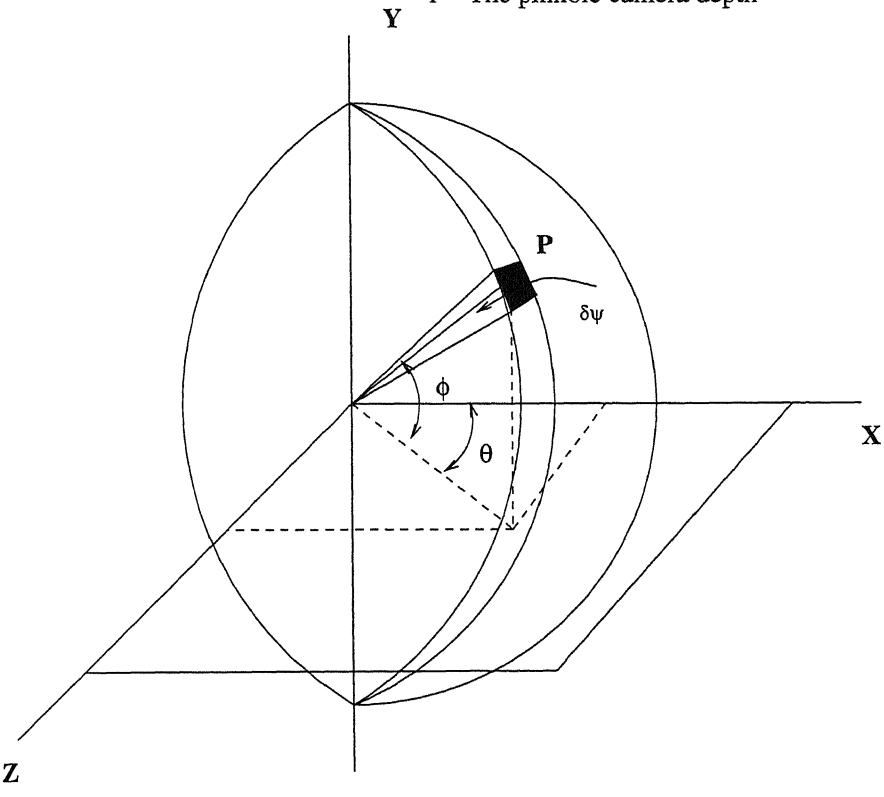


Figure 2.9: *Hand Orientation in Work Space* The shaded region P represents the hand tp position. $\delta\psi$ = solid angle subtended in the particular direction. θ = Azimuth angle. ϕ = Elevation Angle.

$$\delta\Theta = \begin{Bmatrix} \delta\phi \\ \delta\theta \end{Bmatrix} = \begin{bmatrix} 0 & \frac{1}{L'\cos\phi} \\ -\frac{1}{L'\sin\theta\cos\phi} & -\frac{\tan\phi}{L'\tan\theta\cos\phi} \end{bmatrix} \begin{Bmatrix} \delta x \\ \delta y \end{Bmatrix} \quad (2.4)$$

The deviation in the solid angle about the pointing direction is $\delta\psi = \sin\phi\delta\phi\delta\theta$ and thus the final equation can be written as:

$$\|\delta\psi\| = \|f_1\delta x\delta y + f_2\delta y^2\| \quad (2.5)$$

where f_1 and f_2 are given by:

$$\begin{aligned} f_1 &= \frac{\tan\phi}{L'^2\sin\theta\cos\phi} \\ f_2 &= \frac{\tan^2\phi}{L'^2\tan\theta\cos\phi} \end{aligned}$$

As f_1 and f_2 have the cosines of the elevation angle in the denominator, slight variation in pixel values near $\phi = \pm\pi/2$ region can give rise to a large deviation in the pointing direction. The same holds for the azimuth angle $\theta = 0$. These are more clearly shown in figure 2.10.

2.4 Results

In this chapter we discuss some of the implementations and experiences with gesture-based control of real devices

2.4.1 Development of Gesture Sets

As mentioned in the previous chapter, two kinds of manipulative gestures have been developed. Symbolic & Pointing. The latter contains directional information as well as a simple symbolic directive.

Gloves Vs. No Gloves

In the development of symbolic gestures, the first set of gestures utilised the gross motion of the human arm. Depending on the positions of the two hand blobs at different positions of the screen the gesture could be recognised. These gestures involve lesser processing, hence are easier to

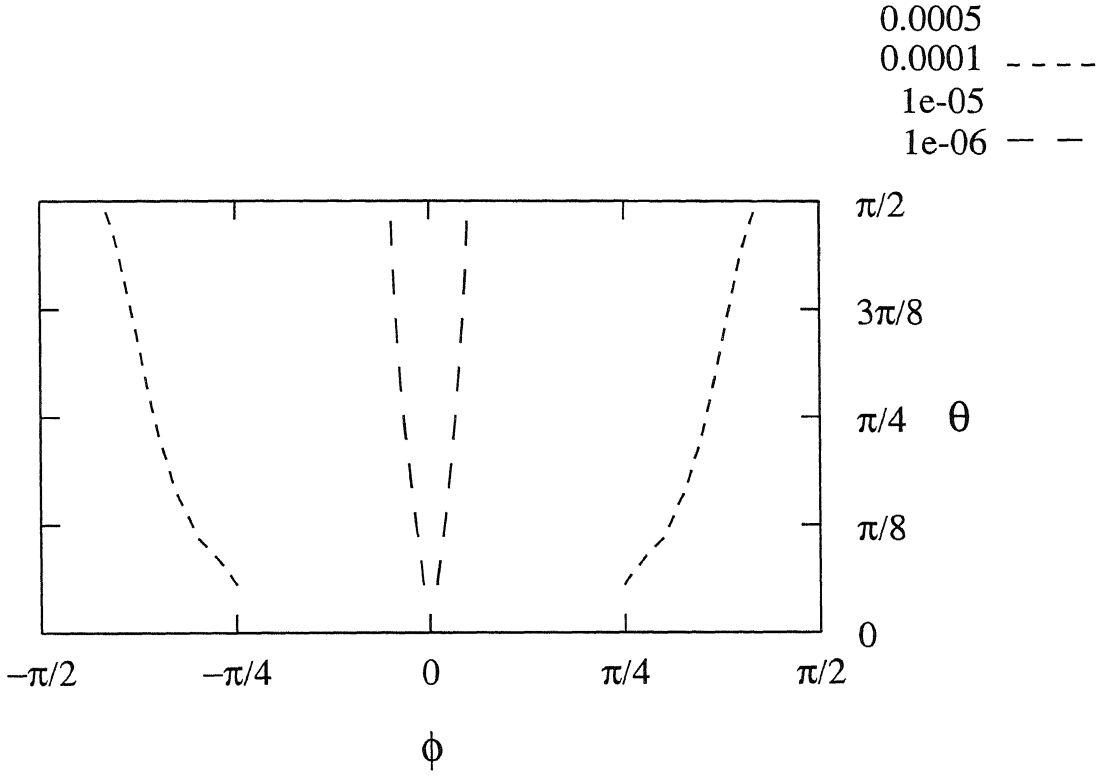


Figure 2.10: *Variation in Solid Angle in a particular direction:* $|\delta x/L'| = |\delta y/L'| = \Delta = 1/400$, alternatively, assuming a hand true length of 400 pixels in the image co-ordinates. Values of the contours of $\delta\psi$ are given in top right corner of the plot

detect, but these use an extra pair of gloves which limits the application. The second category of gestures are dependent purely on features of the hand. A distinction of the hand from the face is also made to make the system fairly robust. The use of the features of the hand to recognise the gesture make them independent to scale.

Given the above argument, the normal opinion will favour the use of a no glove version of the gestures. But, determination of such a step is purely application oriented An off-site operation precludes use of any special equipment, yet use of special clothing can be justified as the operator has little control over the lighting unlike the laboratory setup. Again, if the control of the device is in a factory environment maximum contrast should be created between the background and the foreground. Then the glove version of the symbolic gestures are more stable than the no glove version.

Pointing Gestures & Sensitivity

Pointing gestures are captured using computer vision. These gestures particularly are highly error prone. The sensitivity due to geometry of the work space is described in section 2.3.1. It was observed a shift of an approximate angle by 10 degrees by the user when the hand was pointing towards the camera made the system to recognise the shift by 6.1 degrees. But, same shift of input in the peripheral region led to an observed shift of 4.9 degrees. Though the output shows lesser error when the hand was pointing towards the camera the accuarcy of pointing is not very close to the actual shift. This error is because when the person points towards the camera locating the tip of the index finger is not easy as our algorithm detects the extreme left point of the man's blob in the image as the pointer location which is actually the extreme right point of the fist of the right hand of the man.

2.4.2 Applications of gestures in controlling real devices

Symbolic gestures were used to control a manipulator as well as a mobile robot. The gestures were assumed to be motion directives. No further context information was used in this work.

Figure 2.11 shows the gloves based symbolic gesture based movement of the PUMA-560 manipulator.

Figure 2.12 shows the same set of gestures used for movement of the mobile robot.

Figure 2.13 the gestures without the gloves used for controlling the mobile robots.

Figure 2 14 Pointing gestures tried on PUMA-560

2.4.3 Virtual Applications

As a matter of peripheral interest some of the same gestures has been used to to control graphic simulations shown in figure 2.15 carried out by [10, 11].

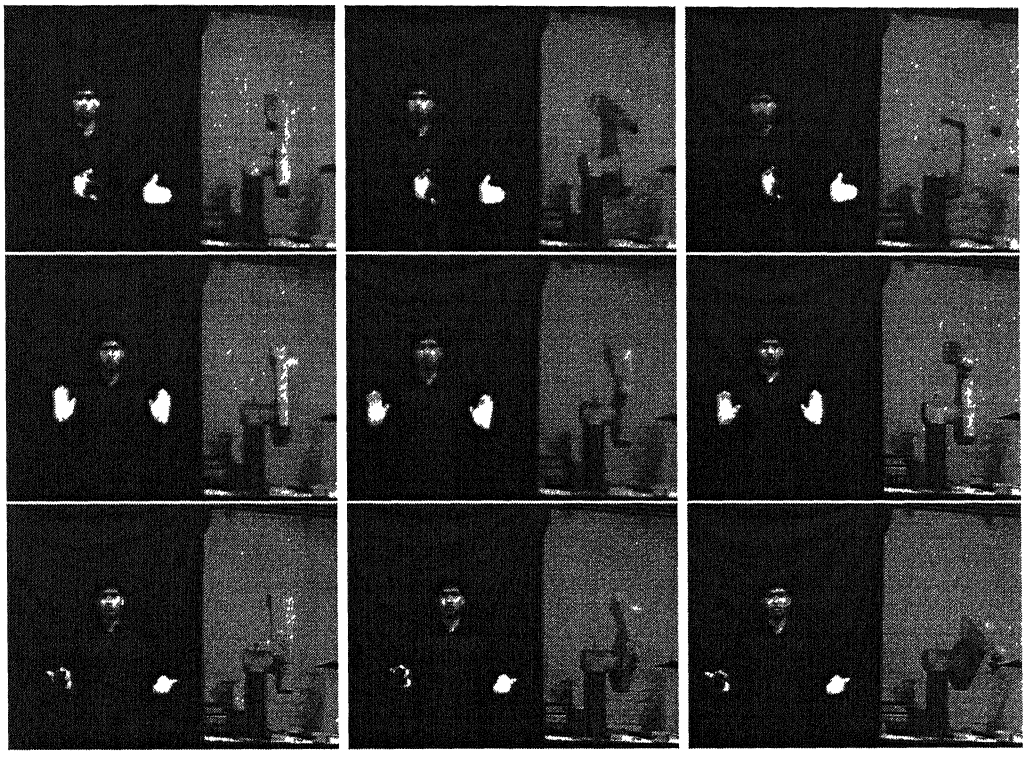


Figure 2.11: *PUMA-560 moving under glove based symbolic gestures:* The PUMA-560 manipulator moving under the gesture commands from the user. Shown in this figure in the top row the Move Left command, middle row the Move Up commands and bottom row the Come Forward gestures. The complete gesture set shown in figure 2.3

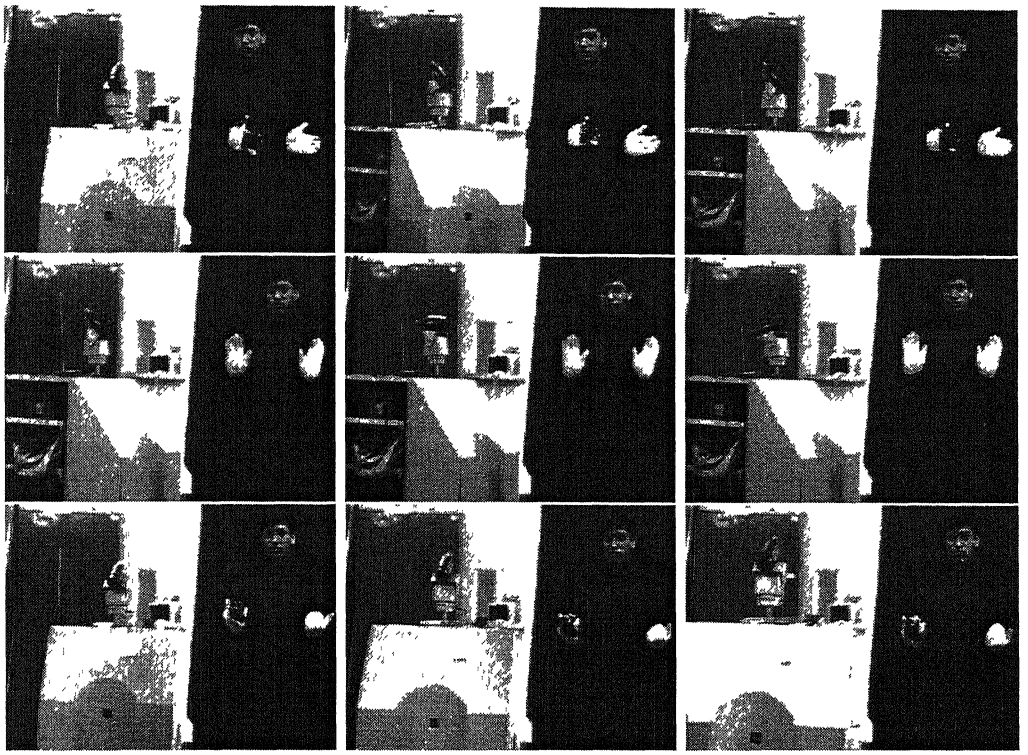


Figure 2.12: *Remotely Operated Mobile Platform (ROMP) moving under glove based symbolic gestures:* The mobile robot ROMP moving under the gesture commands from the user. Shown in this figure in the top row the Move Left command, middle row the Move Up command and bottom row the Come Forward gesture. The complete gesture set is as shown in figure 2.3. Move Left is interpreted as Turn Left and Move Up command is interpreted as Tilt Up of the camera mounted on ROMP. In this figure, other commands not shown and interpreted differently are Move Right as Turn Right and Move Down as Tilt Down for the camera.

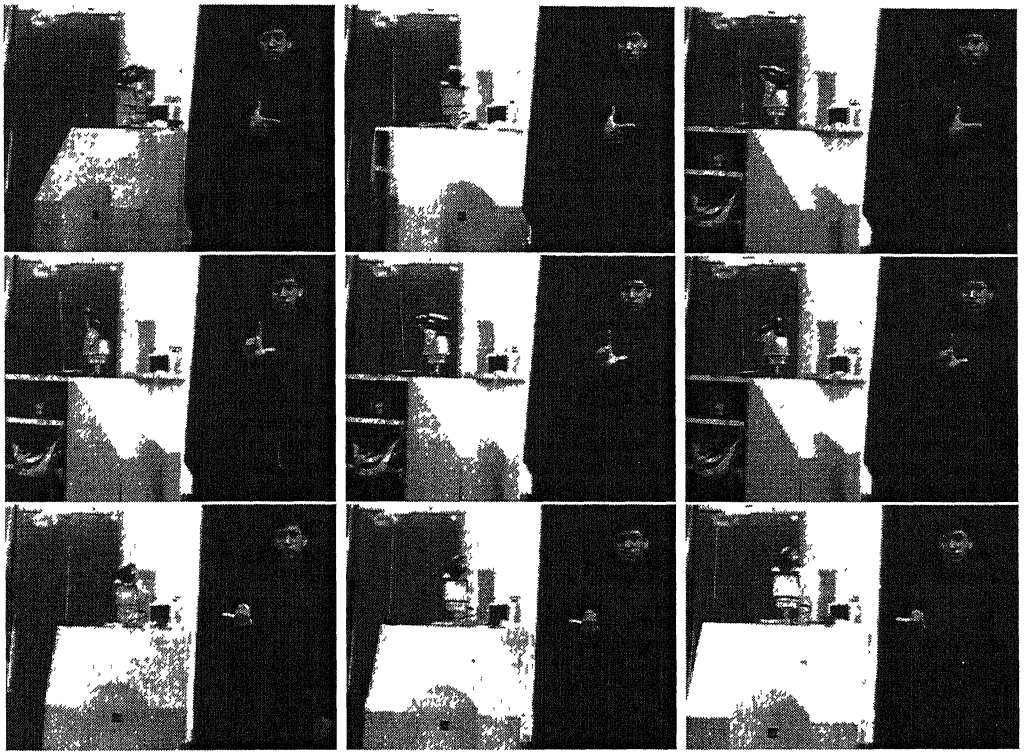


Figure 2.13: *Mobile robot ROMP moving under symbolic gestures without using the gloves:* The mobile robot ROMP moving under the gesture commands from the user. Shown in this figure in the top row the Move Left command, middle row the Move Up commands and bottom row the Come Forward gestures. The complete gesture set shown in figure 2.4. Move Left command is interpreted as Turn Left and Move Up command is interpreted as Tilt Up of the camera mounted on the ROMP. In this figure, other commands not shown and interpreted differently are Move Right as Turn Right and Move Down as Tilt Down for the camera.

Figure 2.14: *PUMA-560 moving under pointing gestures:* The PUMA-560 manipulator moving under the pointing gestures from the user. Shown in the figure three pointing directions and effective motion of the PUMA-560. The motion of the manipulator wrist is with reference to its earlier position.

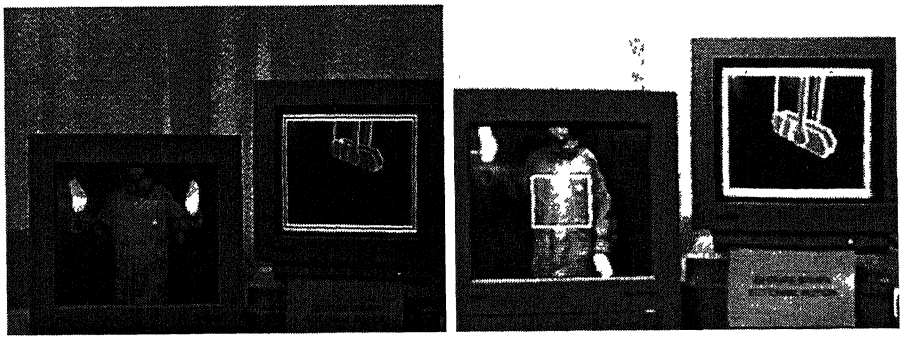


Figure 2.15: Work done by used by Mishra_et_al:1995 using the some of the gestures similar to gestures used by us

Chapter 3

Limitations & Scope of further work

The application of current work on gestures is limited in the following respects. The gesture models used are fixed models which are prone to lighting conditions. In this work we have partially been successful in eliminating such problems by using an auto focus & auto iris lens. The presence of various objects in the background has to a great extent limited its application. The operator's clothing has also created significant problems in gesture recognition. Hence, we have relied on the use of dark clothing and dark background. The following are a few suggested modifications which can put the current work on a sounder footing.

Background Elimination

One of the common background elimination techniques for fixed backgrounds is to grab the background image before the operator enters and to subtract in absolute pixel values the background image from the subsequent images. Thus, the pixel values which are the same in both background and foreground get eliminated. But, this has its limitations in eliminating the light sources in the background which get occluded by the presence of an operator. Again, the reflection patterns in the room change with the presence of an operator. An alternative solution to this problem can be to implement

the following idea:

$$pixel_{final} = \begin{cases} pixel_{curr} - pixel_{back} & \text{if } pixel_{curr} > pixel_{back} \\ 0 & \text{if } |pixel_{curr} - pixel_{back}| < \text{Threshold} \\ pixel_{curr} & \text{otherwise} \end{cases}$$

This could not be tried on our system as Matrox Imaging Library (MIL Version 2), we used doesn't support pixel-wise direct access of the image buffer. Addressing the image buffer through the Image Series Native mode Toolkit gives some flexibility in accessing the VRAM of the image card directly which has not been tried in this work. The alternatives used by different authors are to use equipments for colour processing [26], use of thermograph camera [15], tracking the person in varying background using crude 3-D kinematic models of human being [8].

Dynamic models for gesture recognition

We have implemented the gestures based on fixed human models. Hence, dynamic gesture recognition is not feasible. Learning models handle such situations in a robust way as suggested by [26]. In fact, the Pfinder searches for the human being in the scene based on 2-D point sets on the human body and these points are trained statistically. The model is updated over the period of time. The equipment involves a colour processing system using both blob and contour processing on a SG Indy machine. It tracks the person at 10Hz. A relatively simpler algorithm suggested by [8] rely on tracking parts of human being by subdividing the image of the human being into groups of proximity spaces and connecting them by a relationship based on the person's height. The use of PRISM-3 system with stereo cameras enable the authors to estimate the spatio-temporal shift of each of the Proximity Spaces. These gestures are quite efficient for mobile robot control and the presence of the camera on the mobile robot adds a lot of significance to this work. But, unlike Pfinder [26] it has no method to learn the operator's model. Other approaches suggested by some authors is to

implement parametric state space models based on Hidden Markov Models [3, 20, 21, 22, 25]. Unlike, statistical template matching techniques those which rely on mean parameter and a deviation around the mean or employ correlation match techniques, here the image is considered as a state of pixels Shift from one pixel set to another can be thought of as a transition of state and can be processed as a Markov Process. In these kind of models the state transition probabilities are tuned on the basis of the input(the image) and output(the gesture it represents) to the system. Once the system is trained, the recognition phase utilises these information to recognise the trained gestures. HMMs fall into the broader class of techniques called as Dynamic Time Warping. Another DTW model suggested by [5] utilises learning of qualitative views based on normalised correlation matching. A temporal order to these view models are added to conduct a fast search within these view models to recognise the gesture. The gesture here is a set of qualitative 2-D views. One of the above mentioned techniques can be employed to improve upon the current work in introducing dynamic gesture recognition. [21, 22] have used HMMs in recognising communicative sign languages. Their work can further be extended to implement on manipulative gesture recognition. Dynamic pointing gestures using HMM and polhemus tracking devices has been reported by [25]. In the current work we have used static pointing gestures based on a single camera input. Hence scope is there to improve upon our model and introduce dynamic pointing gestures using HMMs and camera input data.

3.1 Towards building up of a Semi-Autonomous System

In this work, we have developed a simple gesture based teleoperation model and discussed about its application in off-site tele-operation. Gestures are used for free motion path planning and have little scope in compliant mo-

tions because of lack of force feedback. Hence, highly specialised tasks like compliant motions (opening a valve etc. described in Chapter 1.) or collision avoidance are to be transfered to a local controller on the robot which can take independent local level decisions. This impart a supervisory role to the operator and thus reduce the effects of human error in controlling the robot. Figure 3.1 gives a broad overview of an intelligent system which can be tele-guided. Currently, collision avoidance systems have been incorporated in various mobile and map builder robots. But, this field is still in its nascent stages of development. The scope for integration of such devices with better human computer interfaces (like speech and gestures) is still remaining as an open challenge in robotics.

3.1.1 Obstacle Avoidance

This is one of the fundamental requirements for a robot to operate safely in the work environment particularly if it has to run with lesser supervisory interventions by an operator which is quite common in a tele-operated system. The limited feedback the operator gets while controlling such systems are sometimes confusing even for a skilled operator. Obstacle avoidance consists of motion planning and collision avoidance. While motion planning is off-line path planning for computing a collision free path, collision avoidance deals with online modification of the planned motion of the robot [9]. When we talk about local level obstacle avoidance the thrust is towards the building of an collision avoidance system. An example of such a system is given in figure 3.2.

While a lot of emphasis has been given on collision free path planning using sensor based systems, now there is a shift towards using vision as the means of obstacle detection. The following are a few reasons for the shift towards vision. Vision has no distance limitation with the use of wide angled lenses to telephoto lenses. Images are spatially dense and have high resolution. It's totally a passive sensor unlike infrared and other such sensor

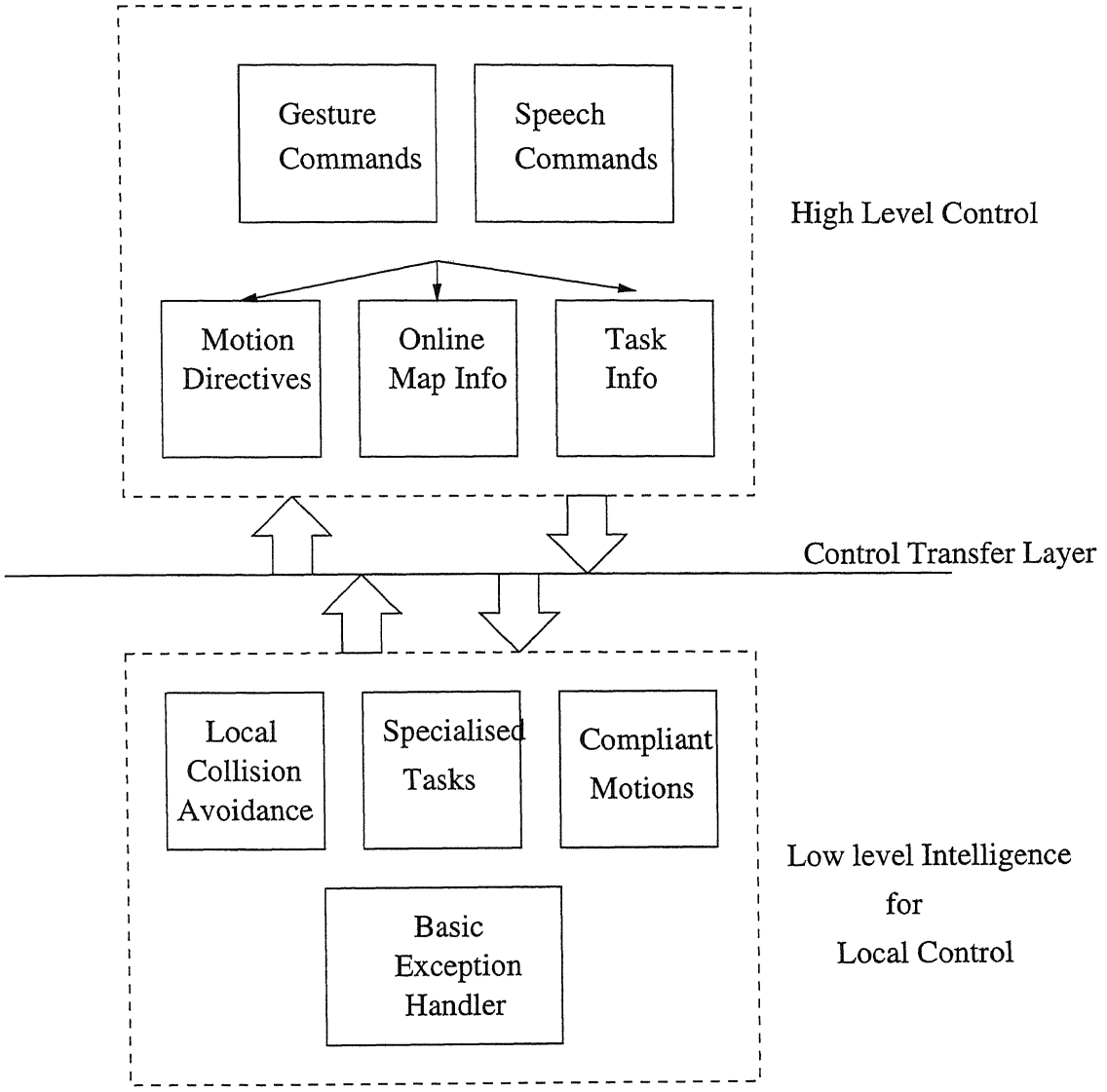


Figure 3 1: *Future of tele-operation:* In future, the tele-operation will shift towards a tele-guidance system. As shown in this figure, a high level set of control commands can be given by gesture and speech. These set of commands will have free space motion commands and guidelines to transfer control to the local system or take the control back to the high level controller (emergency stop). The local on-board control station will have basic collision avoidance modules, or task specific information for compliant motion tasks etc. The basic exception handler in the on-board system should have at least the level of intelligence to recognise an error condition and transfer the control to the operator.

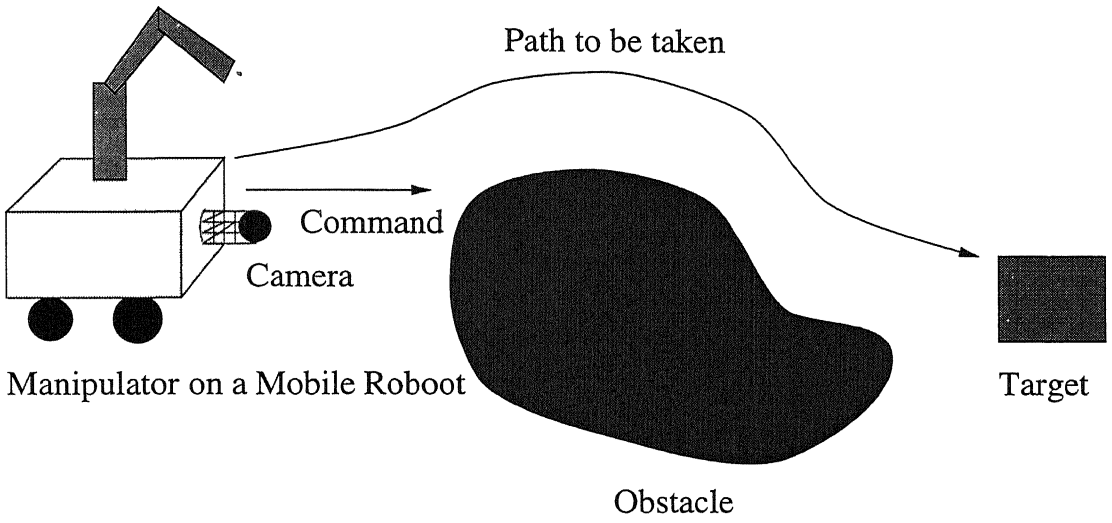


Figure 3.2: *Obstacle avoidance by the local level control:* The move forward command is given by the human operator. Because of the presence of the obstacle on the way, the robot will tend to move in the path shown. Of the two possible option of turning left or right, the robot will move along the least potential field around its vicinity. But, it may so happen that the robot may not able to find the goal and will get stuck in a local minima. Such situations should be handled by the human operator who has the knowledge of the workspace to guide it in the proper direction

which are intrusive to sensitive environments. Obstacle avoidance using optical flow has been reported by [1], where they use the optical flow field divergence suggested by [13] to detect obstacles.

Optical Flow

Two major cues of depth information are stereo vision and motion field analysis. When an object moves in 3-D space the projection of 3-D motion field on the retina of an observer represents a 2-D motion field. When a motion occurs, in an image the only appearable difference is the intensity variation $\Delta E(x,y)$ over the time period t. The optical flow is a vector field describing this intensity change which indicate motion from one feature to other. An example of optical flow is given in figure 3.3.

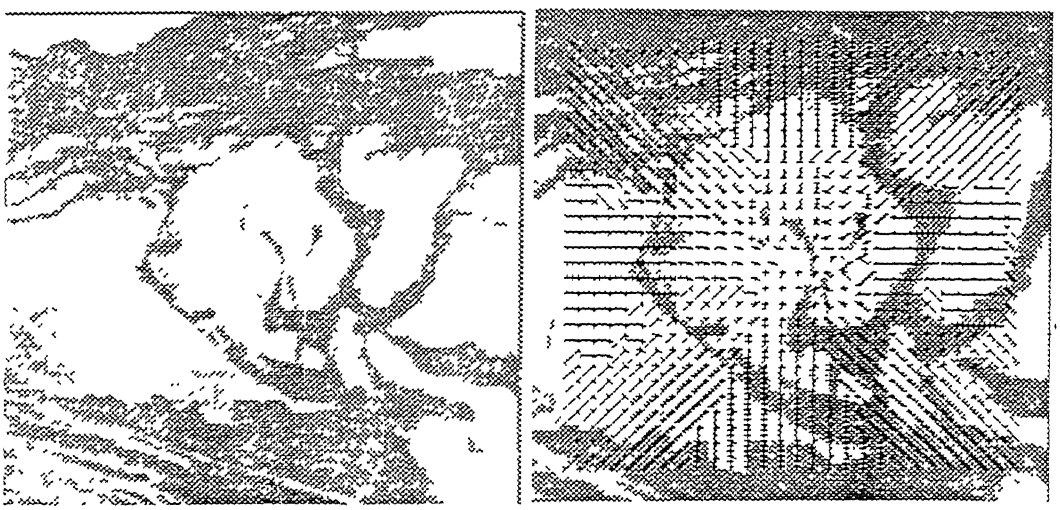


Figure 3.3: *Optical Flow:* Needle diagram showing optical flow magnitude and direction from 1st state to 2nd state. Figure taken from[4]

There are three ways in which optical flow is measured [4]. These are

- Gradient based Optical Flow
- Velocity tuned filter optical flow
- Correlation based optical flow

We will discuss the correlation based optical flow as it is simpler and faster to compute with a linear time algorithm has been proposed by [4]. A brief description of correlation based algorithm has been shown in figure 3 4.

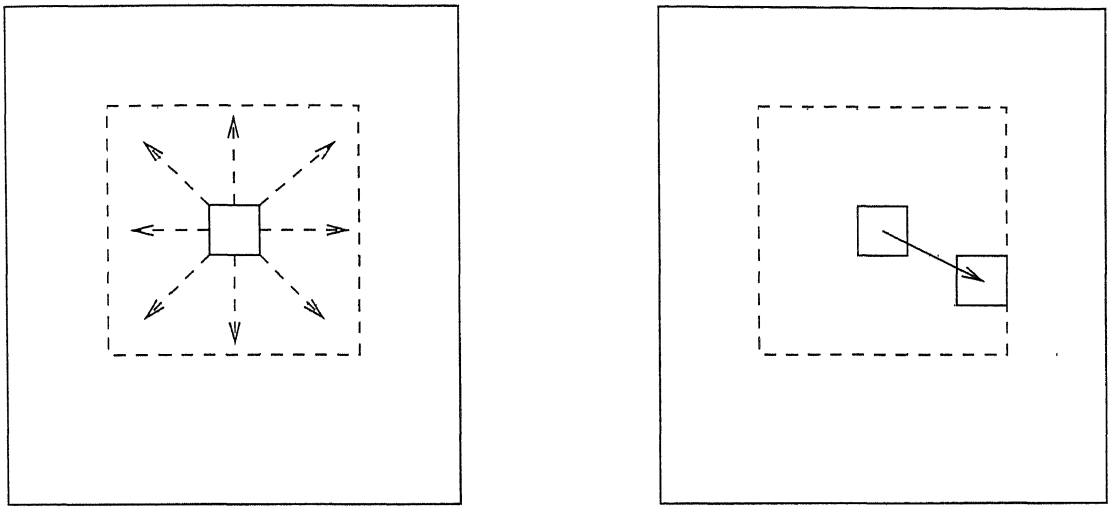


Figure 3.4: *Correlation based optical flow:* As the object moves in the scene there is a shift of pixels from current pixel positions to neighbourhood pixel positions. Assumption taken here is the intensity of the pixels of an object does not change during motion. It is again assumed that within a time step δt a pixel can move from the current position to another pixel exactly η pixels away from its current pixel group position. Thus a possible locations for the pixel can be any of $(2\eta + 1) \times (2\eta + 1)$ possible locations. A winner take all algorithm says that the pixel has moved to that direction where the total correlation match for that pixel is the maximum. Shown in the figure with $\eta = 2$ and which means the pixel can be anywhere of the 25 possible locations in the following time step.

Match strength:[1] In the figure 3.4 we have considered the motion of a

[1]notations and few lines taken from[4]

single pixel. But, studying of single pixel motions in subsequent images can be confusing and may lead to erroneous results as image data has a lot of noise which cannot be eliminated in a single pixel level. The motion of pixel patch of $\nu \times \nu$ centered at [x,y] is studied in practice 3.5. Such a patch (image feature) can lie on any of $(2\eta + 1) * (2\eta + 1)$ possible displacements. The correct motion patch of pixels is determined by simulating motion of the patch for each possible displacement and computing the correlation match for each displacement. If ϕ represents the matching function which returns a value proportional to the match of two given features, then the match strength M(x,y:u,w) for a point [x,y] and displacement (u,w) is calculated by taking the sum of the match values between each pixel in the displacement patch P_ν in the first image and corresponding pixel in the actual patch in the second image:

$$\forall u, w \cdot M(x, y : u, w) = \sum \phi(E_1(i, j) - E_2(i + u, j + w)), (i, j) \epsilon P_\nu (3.1)$$

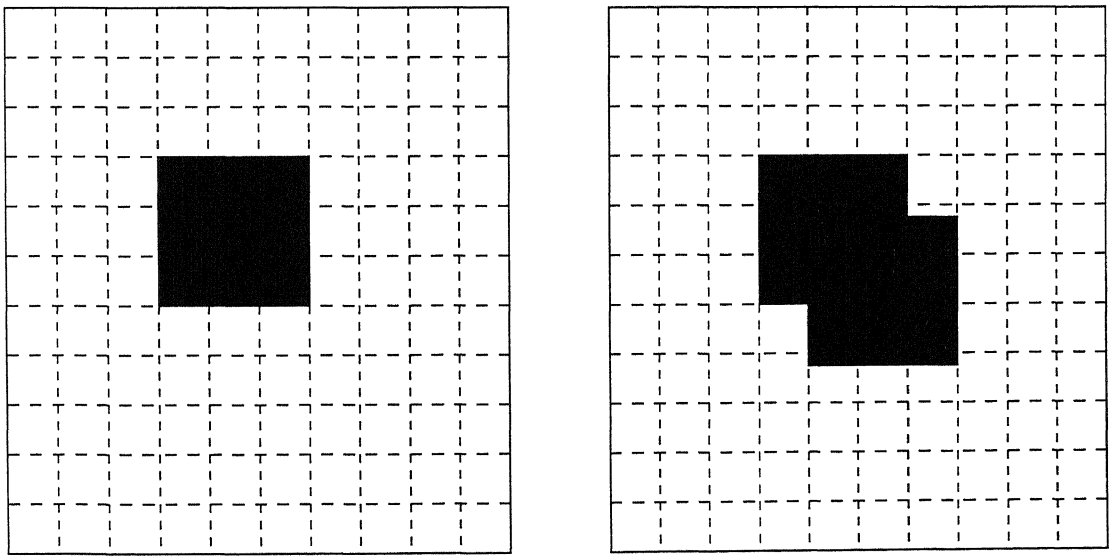


Figure 3.5: *Multiple pixel match strength computation:* In this figure the $\nu = 3$.

A simple choice for ϕ can be to use the absolute value function. The

winner takes all algorithm assumes that the motion has occurred in the direction for the feature where the match strength is the maximum. This means for the choice given for ϕ above the motion will be considered to have occured in the direction where M(x,y:u,v) is minimum.

Linear time optical flow algorithm:[2] We know the velocity is the rate of change of distance. In the scheme described above the search in the space is $O(\eta^2)$. As time steps are linear if we consider a search in the time steps the order of the algorithm is linear time. Hence, if a search is carried out in the time space keeping spatial shift of pixel to a maximum of $\eta = 1$ pixel over a time steps t=1,..,n, then this computation becomes linear time. As a maximum shift of 1 pixel will occur in one of 1,2,..,n time steps sub-pixel motions 3.6 are computed for each time step.

That means to compute the current pixel value optical flow vector upto n previous time step images are required. The match strength described above considers the match strength in the spatial domain comparing current image only with the previous image. But, in case of linear time optical flow measurement the match strength is to be computed on the basis of comparing the maximum of the correlation matches over the n time steps. So, for the spatial shift estimate of η for each time step $(2\eta + 1) * (2\eta + 1)$ searches are to be conducted n times. Thus a total of $n(2\eta + 1) * (2\eta + 1)$ searches are conducted. Keeping $\eta = 1$, this value comes down to $9n$.

3.2 Conclusion

In brief, we have started with studying human gestures, classified them into various gesture classes depending on the intent of their use, temporal variations and levels of usage of human anatomy. We have looked into various kinds of gesture models used by various authors. An off-site gesture based tele-operation paradigm was developed and suitability of this paradigm in

[2] introduced by [4]

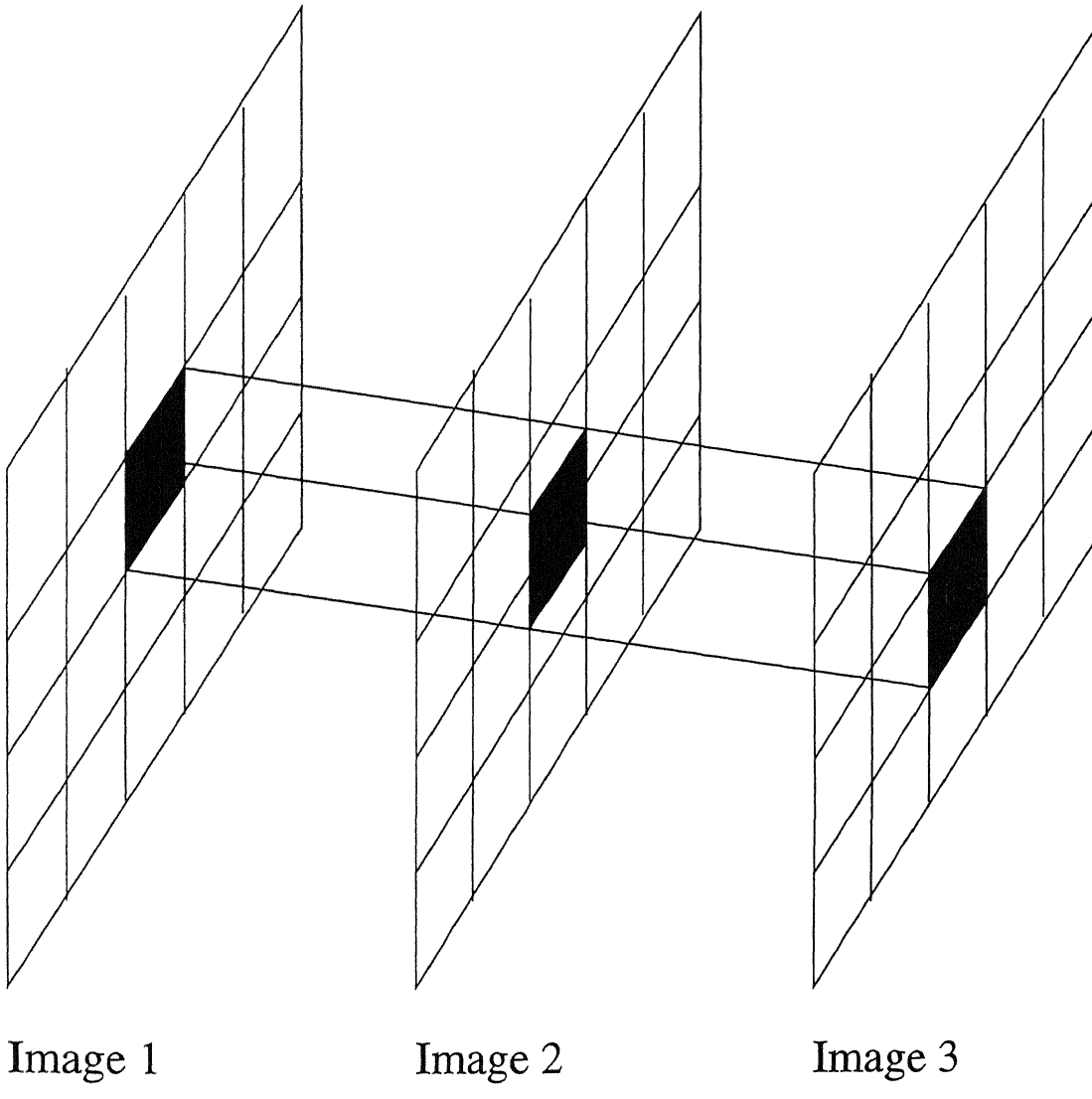


Figure 3.6: Sub-piexel motion:In the image 2 - the shift of the pixel is 1/2 a pixel

tele-operation is discussed.

In the second chapter, we have stressed on developing a position based as well as a shape based symbolic gesture set for real device control. These gesture were used to control a manipulator and a mobile robot. Static pointing gestures were captured using a single camera input. A theoretical estimate of sensitivity of such pointing gestures were discussed. These pointing gestures were later used to control a manipulator.

In the concluding chapter, we have compared our work vis-a-vis work done by other authors and described the future teleoperation systems which will use high level human gesture based control and low level on-site decision making in terms of obstacle avoidance and compliant motion execution. The vision based obstacle avoidance using optical flow was described in some details. Particularly, the correlation based optical flow estimation was discussed with greater details. In a nutshell, we have integrated work done by various authors and presented it in a condensed format in this work.

Bibliography

[1] M. Brand and I. Essa. Causal analysis for visual gesture understanding. Technical Report TR-327, MIT Media Lab., Learning and Common Sense Group(1) Perceptual Computing Group(2) MIT Media Lab. Cambridge, MA 02139, USA, November 1995. Appeared in the Proc. of AAAI Fall'95 Symposium on Computational Models for Integrating language and Vision.

[2] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. Perceptual Computing/Learning and Common Sense TR-407, MIT Media Lab, Vision and Modeling Group MIT Media Lab Cambridge, MA 02139 USA, Nov 1996.

[3] T Camus *Real-Time Optical Flow.* PhD thesis, Brown University, September 1994.

[4] T. Camus, D. Coombs, M. Herman, and T. Hong. Real-time single-workstation obstacle avoidance using only wide-field flow divergence. In *13th ICPR Application and Robotic System, Vienna, Austria*, August 1996.

[5] T Darell and A Pentland Space-time gestures. *CVPR/NYC*, June 1993.

[6] E. Hunter, J Schlenzig, and Ramesh Jain. Posture estimation in reduced-model gesture input systems. In *Intl.*

Workshop on Applications of Face and Gesture Recognition, [http://vision.ucsd.edu/papers/ zurich95.ps.gz,, 1995.

[7] A. Katkere, E. Hunter, D. Kuramura, J Schlenzig, S. Moezzi, and R. Jain. Robogest: Telepresence using hand gestures. Technical Report VCL-94-104, Visual Computing Laboratory, University of California, San Diego,, December 1995. also availabe at: [http://vision.ucsd.edu/papers/ robogest-iros.ps.gz].

[8] D. Kortenkamp, E. Huber, and P. Bonasso. Recognising and interpreting gestures on a mobile robot. In AAAI, editor, *Thirteenth national conference on Artificial Intelligence, AAAI*, August 1996.

[9] T. Leo, S. Longhi, and R. Zulli. On-line collision-avoidance for a robotic assistance system. In P. Kopacek, editor, *Human-oriented design of advanced robotic systems, Postscript volume from the IFAC, Workshop, Vienna. Austria*, pages 81–86, Dipartimento di Elettronica ed Automatica. Università di Ancona-Italy, September 1995.

[10] N K. Mishra, M.P Singh, T.V. Prasannaa, B.K. Birla, D. Vidhani, A.N. Lal, and A Mukerjee. Gesture based user interfaces: an alternative model for virtual reality. In Tata McGraw Hill & Co., editor, *Proceedings of the International Conference on Cognitive Systems, ICCS-95, New Delhi*, December 1995.

[11] N.K. Mishra, M.P. Singh, T V. Prasannaa, B.K. Birla, D. Vidhani, A.N. Lal, and A. Mukerjee. Experiments in gesture based user interfaces. In *Conference of the Computer Society of India, (CSI-96),Bangalore, India, October 1996.*, October 1996.

[12] A. Mukerjee and S. K. Dash. Off-site tele-operation using gestures. In M. Vidyasagar, editor, *Proceedings ISIRS-98*, Bangalore, January 1998. This work is also included in the Master's thesis of Sambit Kumar Dash.

[13] R Nelson and J. Aloimonos. Obstacle avoidance using flow field divergence. 11(10):1102 1106, October 1989.

[14] D J Oborne *Ergonomics at Work.* John Wiley & Sons Ltd., Dept. of Psychology, University College of Swansea, 1 edition, 1982.

[15] J. Ohya and K Sengupta. Generating virtual environments for human communications - virtual metamorphosis system and novel view generation. In *Computer Vision for Virtual Reality Based Human Communication (CVVRHC)*, pages 43-50, January 1998.

[16] V. I. Pavlovic, R. Sharma, and T. S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 19(7), July 1997.

[17] P.K. Pook and D. Ballard. Tele-assistance: contextual guidance for autonomous manipulation. In *AAAI-94*, pages 1291–1296, {pook,dana}@cs.rochester.edu, 1994.

[18] Jim Rehg and Takeo Kanade. Digiteyes: Vision-based human hand tracking Technical Report CMU TR CMU-CS-93-220, CMU, `ftp://reports.adm.cs.cmu.edu/usr/anon/1993/CMU-CS-93-220.ps.Z`, May 1993.

[19] Jim Rehg and Takeo Kanade. Visual tracking of self-occluded articulated objects. Technical Report CMU TR CMU-CS-94-224, CMU, `ftp://reports.adm.cs.cmu.edu/usr/anon/1993/CMU-CS-94-224.ps.Z`, 1994.

[20] J. Schlenzig, E. Hunter, and R Jain. Recursive identification of gesture inputs using hidden markov models In IEEE Computer Society Press, editor, *Proceedings of the Second IEEE Workshop on Applications of Computer Vision*, pages 187-194, December 1994. Also available on [`http://vision.ucsd.edu/ papers/WACV94.ps`].

[21] T. Starner. Visual recognition of american sign language using hidden markov models. Master's thesis, MIT, February 1995. Also VISMOD TR-316 [http://www-white.media.mit.edu/vismod/people/publications/publications.html].

[22] T. Starner and A. Pentland. Visual recognition of american sign language using hidden markov models. In *International Workshop on Automatic Face and Gesture Recognition*, Zurich, Switzerland, 1995. also available on: [http://www-white.media.mit.edu/vismod/people/publications/publications.html].

[23] D. J. Sturman and D. Zeltzer. A survey of glove-based input. *IEEE Computer Graphics & Applications*, pages 30–39, January 1994.

[24] A. Wilson and A. Bobick. Using configuration states for the representation and recognition of gesture. Technical Report TR-308, MIT Media Lab , 1995. Also Published abbrev. version in ICCV' 95.

[25] A. Wilson and A Bobick. Recognition and interpretation of parametric gestures. Technical Report TR-421, MIT Media Lab., 1997. Published in ICCV 1998.

[26] C. R. Wren, A. Azerbayejani, T. Darell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Trans. on Pattern Recognition and Machine Intelligence*. July 1997.